

Research Article

River Flow Lane Detection and Kalman Filtering-Based B-Spline Lane Tracking

King Hann Lim,¹ Kah Phooi Seng,² and Li-Minn Ang³

¹Electrical and Computer Department, School of Engineering, Curtin University Sarawak, CDT 250, Sarawak, 98009 Miri, Malaysia

²School of Computer Technology, Sunway University, No. 5, Jalan Universiti, Bandar Sunway, Selangor, 46150 Petaling Jaya, Malaysia

³Centre for Communications Engineering Research, Edith Cowan University, Joondalup, WA 6027, Australia

Correspondence should be addressed to King Hann Lim, glkhann@curtin.edu.my

Received 27 March 2012; Accepted 26 September 2012

Academic Editor: T. A. Gulliver

Copyright © 2012 King Hann Lim et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A novel lane detection technique using adaptive line segment and river flow method is proposed in this paper to estimate driving lane edges. A Kalman filtering-based B-spline tracking model is also presented to quickly predict lane boundaries in consecutive frames. Firstly, sky region and road shadows are removed by applying a regional dividing method and road region analysis, respectively. Next, the change of lane orientation is monitored in order to define an adaptive line segment separating the region into near and far fields. In the near field, a 1D Hough transform is used to approximate a pair of lane boundaries. Subsequently, river flow method is applied to obtain lane curvature in the far field. Once the lane boundaries are detected, a B-spline mathematical model is updated using a Kalman filter to continuously track the road edges. Simulation results show that the proposed lane detection and tracking method has good performance with low complexity.

1. Introduction

Automation of vehicle driving is being developed rapidly nowadays due to the vast growth of driver assistance systems (DASs) [1]. In conjunction with the development of low-cost optical sensors and high-speed microprocessors, vision-based DASs become popular in the vehicular area to detect apparent imaging cues from various road scenes for visual analysis and therefore warn a driver of an approaching danger and simultaneously perform autonomous control to the vehicle's driving. Of all fatal errors happened, driver's inattention and wrong driving decisions making are the main factors of severe crashes and casualties on road [2]. The deviation of a vehicle from its path without a signal indication has threatened the nearby moving vehicles. As a consequence, vision-based lane detection and tracking system becomes an important mechanism in vehicular autonomous technology to alert a driver about road physical geometry, the position of the vehicle on the road, and the direction in which the vehicle is heading [3].

In the last few decades, a lot of vision-based lane detection and tracking techniques [4–8] have been developed in order to automatically allocate the lane boundaries in a variety of environmental conditions. It can broadly be divided into three major categories, that is, region-based method, feature-driven method, and model-driven method. Region-based method [9–14] basically classifies the road and non-road pixels using color or texture information. Although it has simple algorithm, it may suffer from color inconstancy and illumination problem. Feature-driven method [15–18] extracts the significant features such as lane markings from road pixels to identify the lane edges. This method is highly dependent on feature detection methods such as edge detection which are sensitive to occlusion, shadow, or other noises. On the other hand, model-driven method builds a mathematical function such as linear-parabolic [19, 20], hyperbola [21, 22], or spline-based [23, 24] methods to mimic the lane geometry. Due to its comprehensive learning and curvature flexibility, this method has been widely used in the lane detection and tracking system.

As stated in [19], Jung and Kelber manually cropped the effective road region to obtain the possible lane edges detection. At the same time, they applied a fixed threshold to split near- and far-field segment for the lines prediction using linear-parabolic model. On the other hand, Wang et al. [24] have proposed a lane detection and tracking technique using B-snake lane model, which measures dual external forces for generic lane boundary or marking. Initially, lane boundaries are detected using Canny/Hough estimation of vanishing points (CHEVPs). It is followed by constructing B-snake external force field for lane detection iteratively based on gradient vector flow (GVF) [25]. Nevertheless, the above-mentioned methods have encountered some flaws in the process. Manual cropping to obtain the effective lane region may not be an efficient way in the automation. In addition, fixed near-far-field threshold or segmented lines are not always applicable to all on-road conditions to determine the edges. Moreover, edge detector and Hough transform (HT) may easily be affected by shadow cast or weather change. The lane boundaries in the far-field range would be gradually undetectable using HT. Furthermore, CHEVP method is sensitive to numerous thresholding parameters initialization. Significant numbers of iterations are required to obtain GVF in the lane-tracking process.

Motivated by the above-mentioned problems, a new system composed of lane detection and tracking is presented in Figure 1. Horizon localization is first applied to a set of traffic scene image sequence to automatically segment the sky and the road region. Road region is then analyzed to further separate nonroad and road pixels. Subsequently, an adaptive line segment is computed using multiple edge distribution functions to monitor the change of road geometry. The portion below the adaptive line threshold is estimated with 1D HT method, while the upper part is determined using a low-processing method with the concept of river flow topology. After the lane edges are successfully located, they are passed over to lane tracking for reducing the computational time. Possible edge scanning is applied to seek for the nearby lane edges with respect to the estimated lane line. Control points are determined in order to construct a B-spline lane model. In assistance with Kalman filtering, B-spline control points are updated and predicted for the following frame's lane curve.

The proposed lane detection and tracking method has offered some advantages over the method [19, 24]. The performance of edge detection and HT are always distorted by shadow effect. Therefore, a regional dividing line is first applied to discard disturbance from sky region. The elimination of shadow effect is achieved by using an adaptive statistical method. Instead of using fix line to segment near and far-field, an adaptive line segment is proposed to monitor the change of angles along the lane boundaries. The concept of river flow is proposed for the lane detection system to follow the road in the far-field region. Furthermore, Kalman filter plays a twofold-role: (i) to correct B-spline control points for the current image frame and (ii) to predict the lane model for the consecutive frames. Unlike [24], less parameter tuning or thresholding values are required in the proposed system. Moreover, it has no camera

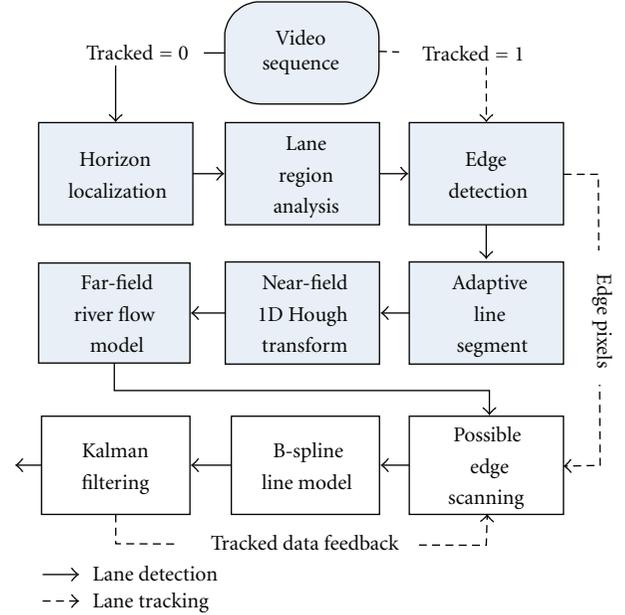


FIGURE 1: Block diagram of the proposed lane detection and tracking system.

parameters involved in the system determination. Overall, it gives a better performance with promising computational speed. This paper is organized as follows. Section 2 discusses the proposed lane detection, and Section 3 explains the Kalman filtering-based B-spline lane tracking technique. Some simulation results are shown in Section 4 and followed by conclusion and future works.

2. Lane Detection

Lane detection is a crucial task to estimate the left-right edges of driving path on a traffic scene automatically. In this section, four-stage lane boundary detection is proposed, that is, (i) horizon localization, (ii) lane region analysis, (iii) adaptive line segment, and (iv) river flow model. In the traffic scene, road region is the main focus for lane detection. First, horizon localization splits the traffic scene into sky and road region. Then, lane region analysis is applied adaptively based on the surrounding environment in order to remove mostly road pixels and keep the lane mark pixels. Adaptive line segment is therefore used to analyze the road edge curvature and separate the road region into near and far-field. In the near-field, a 1D HT is applied to draw a near-field line. At the same time, a river flow method is applied to obtain far-field edges which are hardly to be estimated using common line detection.

2.1. Horizon Localization. To eliminate the disturbances from sky segment, horizon localization [26] is performed to partition an $M \times N$ image into sky and road region, whereas M and N are the image row and column, respectively. First, a minimum pixel value filter, with a 3×3 mask, is applied on the image $I(x, y)$ as depicted in Figure 2(a) to enlarge the effect of low intensity around the horizon line. Subsequently,

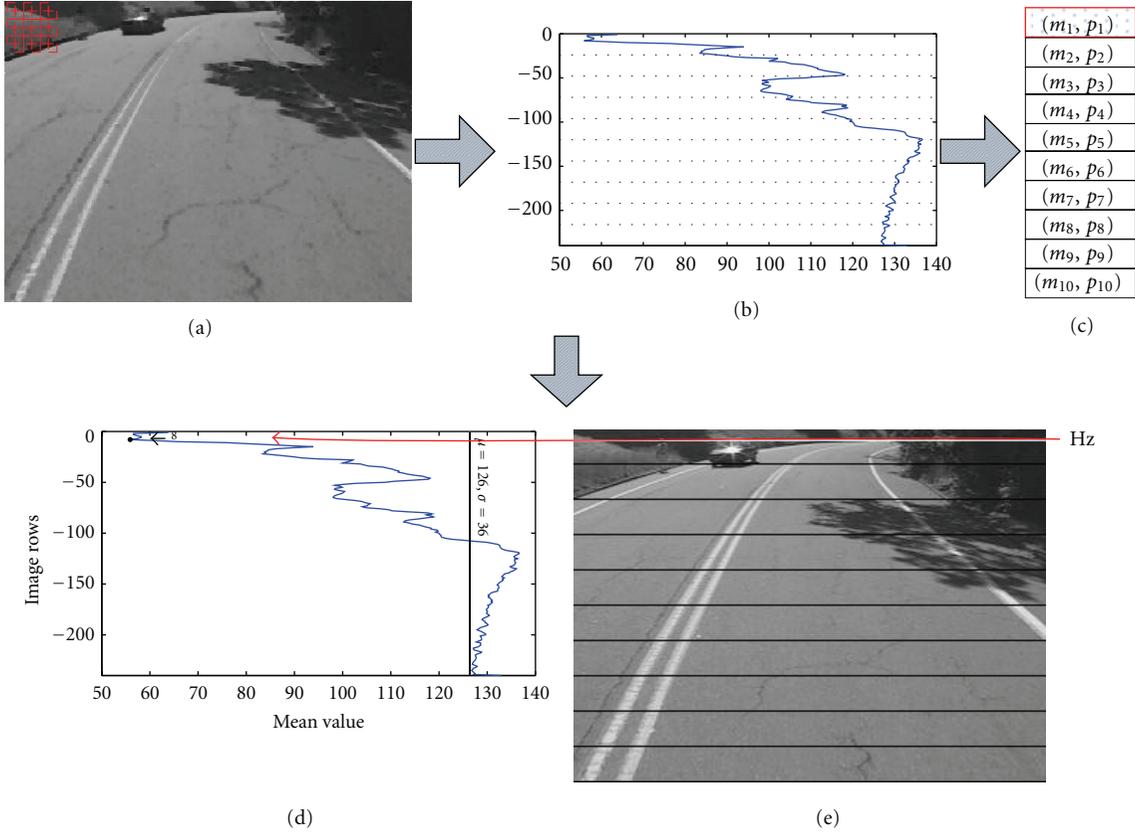


FIGURE 2: (a) Minimum pixel value filter, (b) vertical means distribution segmentation, (c) regional minima, (d) selection of the horizon line threshold, and (e) separation of the sky and road region.

a vertical mean distribution is computed as plotted in Figure 2(b) by averaging every row of gray values on the blurry image. Instead of searching for the first minimum value along the upper curve, the plot curve is then divided into β segments to obtain minima, whereas β represents the number of dividing segments. In this context, β is chosen to be 10 throughout the experiments. All regional minima are recorded to the subset A as follows:

$$A = \{(m_i, p_i) : m_i \in [0, 255] \mid p_i \in [1, M] \mid 1 \leq i \leq \beta\}, \quad (1)$$

where m_i is the magnitude of row pixel mean, and p_i is the row index where the minimum point occurred as shown in Figure 2(c). Naturally, sky region always appears on top of the road image. Therefore, m_1 is taken as a reference minimum point for condition comparison to prevent local minima occurred in the sky portion. Additionally, mean value (μ) of the entire image is calculated to determine an overall change in intensity. The regional minimum point search for the horizon localization is written as follows:

$$Hz = \begin{cases} p_1, & \text{if } (m_1 < m_{i+1}) \cap (m_1 < \mu) \cap (p_1 > \varepsilon_{Hz}), \\ p_i, & \text{else if } [(m_i < m_{i-1}) \cup |m_{i-1} - m_i| < \Delta m] \\ & \cdots \cap (m_i < m_{i+1}) \cap (m_i < m_1), \\ & \cdots \cap (m_i < \mu) \cap |m_{i+1} - m_i| > \Delta m, \\ \lambda, & \text{if others,} \end{cases} \quad (2)$$

where ε_{Hz} is a small integer to prevent a sudden drop from the top of image; Δm is the minor variation of mean value change, whereas $\Delta m = \pm 2$ intensity value, and λ is a user-defined value in case the minimum point cannot be found in the plot. As illustrated in Figure 2(d), the adaptive value of regional dividing line is obtained throughout the regional minimum search, whereas $Hz \in A$ is denoted as the horizon line. The notion of getting the horizon line is because the sky usually possesses higher intensity than road pixels, and it might have a big difference of intensity as the sky pixels approach the ground. Nevertheless, the horizon line often happens at neither the global minimum nor the first local minimum appeared from the upper curve. Hence, a regional minimum search by obtaining minimum points from the segments is proposed to ensure the correct localization for the horizon line dividing the sky and road region, accurately. In Figure 2(e), the horizon line threshold is applied to separate the sky and road region and a road image R_{roi} is generated where all vertical coordinates below the Hz value are discarded.

2.2. Lane Region Analysis. Lane region analysis is performed with an adaptive road intensity range to further classify road and nonroad pixels with regards to the variation of surrounding environment. The lane region analysis steps are described as follows.

Step 1. Select κ rows of pixels for lane region analysis where κ is the number of pixel rows to be selected. The selected rows are started at δ number of rows from the bottom of road image to avoid the likely existence of interior part of a vehicle at the image edge.

Step 2. The intensity voting scheme is carried out on every selected row and each maximum vote of selected row, is defined as v_i , while g_i is the grey value whereas the maximum vote occurs, assuming that

$$B = \{(g_i, v_i) : g \in [0, 255] \mid v \in \mathfrak{R} \mid 1 \leq i \leq \kappa\}. \quad (3)$$

Subset B contains all maximum votes and gray value pixels for i th selected row. Hence, the vote, noted as v_m , and the greatest gray value threshold, noted as g_m , are selected as

$$(g_m, v_m) = \max(B), \quad (4)$$

where $\max(B)$ is a function to get a maximum value from the set B . In addition, the most frequent gray level of the selected region is recorded as g_s , where the global maximum vote occurs for the entire selected pixels. The standard deviation of selected rows is marked as σ_s .

Step 3. Define the adaptive road intensity range as $[g_s - \sigma_s, g_m + \sigma_s]$. Pixels that fall within the range are denoted as possible road pixels, and a binary map (R_{bin}) is formed as depicted in Figure 3(a). This region of interest could be further analyzed to investigate the road conditions. However, in our concern, lane marks are main features to identify the direction of lane flow. Only high intensity values are considered as the lane marks, whereas the pixel values being greater than $g_m + \sigma_s$ are denoted in R_{plm} as "1". This processing step may get rid of shadow problem since shadow is usually in low intensity.

Step 4. By summing up each row of R_{plm} , the values being greater than a threshold (T_1) are discarded to remove the possible high intensity of a vehicle at the frontal view of the image. T_1 is obtained by averaging nonzero rows of R_{plm} .

Step 5. Finally, a difference map (D_{map}) is generated by multiplying R_{roi} and R_{plm} maps. The remaining binary pixels are possible lane mark pixels as shown in Figure 3(b).

2.3. Adaptive Line Segment. Lane markings are the salient features on the road surface, and they are often used to define the boundaries of road region. Initially, the gradient magnitude (∇D_{map}) and the orientation (θ) are denoted as

$$\begin{aligned} |\nabla D_{\text{map}}| &\approx |D_x| + |D_y|, \\ \theta &= \tan^{-1}\left(\frac{D_y}{D_x}\right), \end{aligned} \quad (5)$$

where D_x is the horizontal edge map, and D_y is the vertical edge map. In order to monitor the changes of lane direction, an adaptive line segment is proposed to split the near- and far-field regions using edge distribution functions (EDFs) [19].



(a)



(b)

FIGURE 3: (a) Extracted road region using the adaptive road intensity range; (b) remaining binary pixels are the possible lane markings.

The edge map is first partitioned into small segments for every T_2 rows, whereas T_2 is the number of rows to be grouped for each partition. Multiple EDFs are applied to these partitions to observe the local change of lane orientation based on the strong edges as denoted in Figures 4(a)–4(c). With reference to the gradient map and its corresponding orientation, multiple EDFs are constructed, with its x -axis is the orientation in the range of, $[-90^\circ; 90^\circ]$, and its y -axis is the accumulated gradient value of each orientation bin. The maximum peak acquired on the negative and positive angles denotes the right and left boundary angles, respectively. Peak values that go below a certain threshold T_3 are discarded, whereas T_3 is the mean EDF value of each partition. As shown in Figure 4(c), there is no detection for right angle since it has no significant edge points existed in the image illustrated in Figure 4(d).

Subsequently, EDFs' grouping is done by joining those angles that are approximately equal into the same array. They are grouped from bottom to top based on the difference of each partition to its previous partition within $\pm 5^\circ$. The observation of orientation variation is that it will have an instant change of lane geometry when it comes to the far field. Assume that there are n th groups of lane orientation. If $n = 1$, it is a linear lane boundary. If $n \geq 2$, $(n - 1)$ th of EDF bins are combined to obtain a global lane orientation (θ_{nf}) for near field, while the n th group refers to the far-field orientation (θ_{ff}). Although shadows deviate the lane edges,

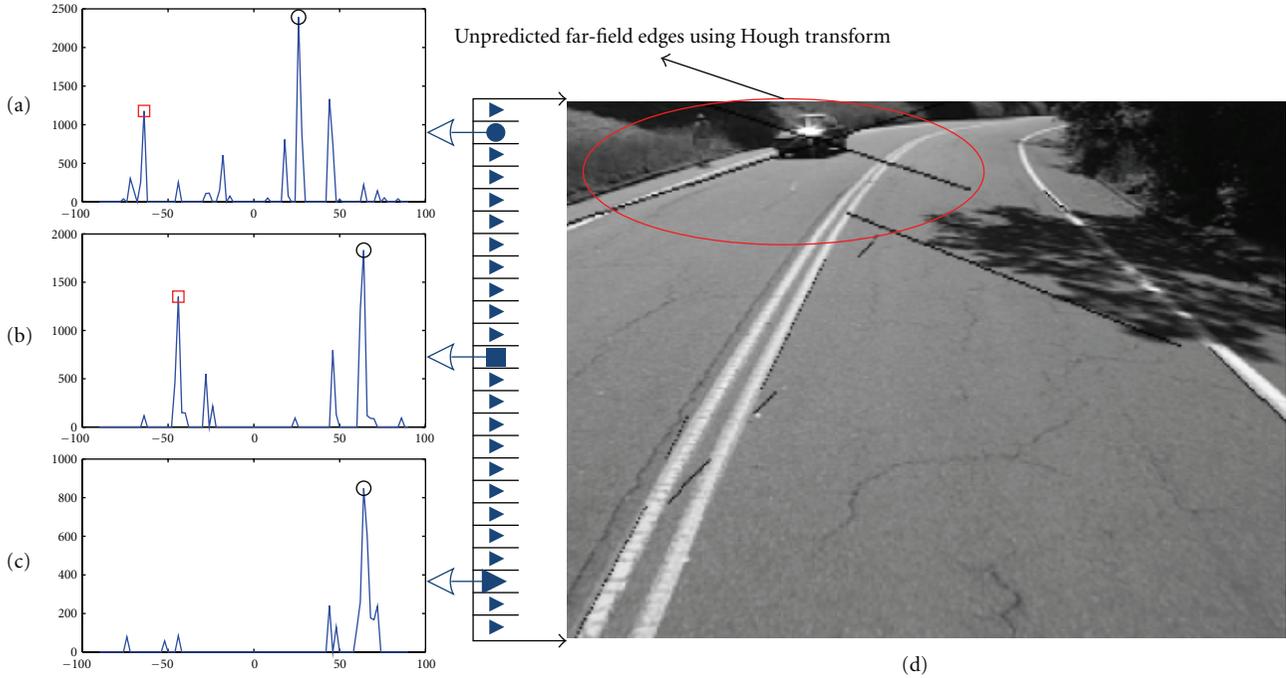


FIGURE 4: (a)–(c) Multiple EDFs on segmented regions, (d) results of lane detection after EDF partitions.

other partitions of EDF with more frequent angle value may correct the orientation of the road path. However, the angle θ_{ff} may not be successfully estimated for the far-field edges because the left-right boundaries may deflect into the same direction.

Eventually, a 1D-weighted gradient HT [19] is applied to near field to determine the lane’s radius $r(\theta_{nf})$ based on $r(\theta_{nf}) = x_o \cos \theta_{nf} + y_o \sin \theta_{nf}$, with known θ_{nf} values. The voting bins for each radius are accumulated with the gradient edge values, and the maximum vote is selected. Figure 5 demonstrates the left-right lane boundaries in the near field constructed by the measured angles and radii.

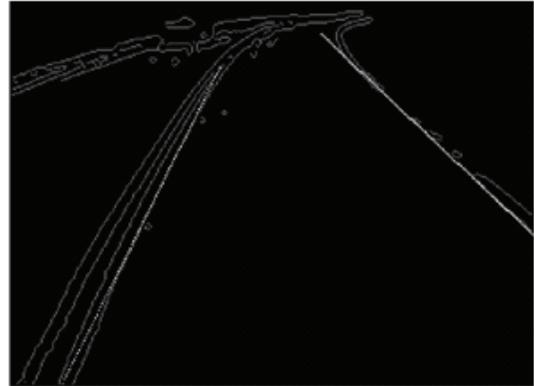


FIGURE 5: Detected lane boundaries by combining $(n - 1)$ th EDF voting bins.

2.4. River Flow Edge Detection. At the far-field region, lane edges flow is hard to be estimated using HT because the lane geometry has become unpredictable in the road scene. As noticed in Figure 4(d), the circle points out the failure of far-field edges detection using EDF and HT. This is because far-field lane edges may have irregular road curvature, and both lane edges may turn into the same direction. Two peaks of angles may fall onto the same side of EDF plot. Therefore, a river flow method is proposed to handle the far-field edge detection. From another perspective, lane flow detection has the same topology of a river flow. When there is an existing path, river will flow along the path. In this context, the path refers to the strength of edges for the lane boundaries. It starts flowing from the closest edges to the adaptive line segment and continues on to the vanishing point regarding to the connectivity and the strongest edge points as shown in Figure 6. With the 3×2 mask provided in Figure 6(b), the selection of the edge will be based on the surrounded large neighboring edge pixel clockwise.

In Figure 6(c), white pixels represent the estimated line from near-field region, while grey pixels represent the edge pixels given by the edge operator. It stops when it has no connectivity or flows in reverse direction.

Assuming the pixels map shown in Figure 7 is an image after edge operator, the river flow is operated in such a manner: check for the nearby high edge pixels and link up all the possible high edge values. The edge pixel having higher value than its neighboring pixels constructs the flow pathway. Initially, the starting flow point has to be allocated in the image before start flowing in the map. In this context, the starting point is allocated by using the previous near-field edge detection technique. As we know, the salient edge points would have higher intensity value than nonsalient pixels. By

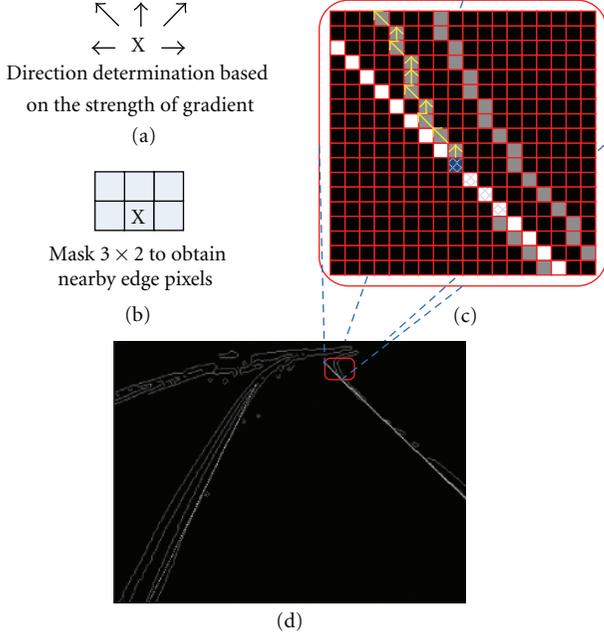


FIGURE 6: The concept of river flow model applied to edge map to detect the most significant edges in the far-field region.

moving the 3×2 mask upwards as shown in the Figure 7, the maximum neighboring pixel is chosen in a clockwise manner indicating the connected path. The flowing process might halt if two or more same higher pixel values existed in the next flow or there is no connectivity detected in the next pixel. Due to the application of lane detection, reversion of the edge point flow is prohibited. The lane flow is either moving forward or moving in the same row of current detected pixel. Finally, the lane edges for left and right road boundaries are detected for lane tracking system.

3. Lane Tracking

After the lane detection is generated in Figure 8, lane tracking system is implemented to restrict the edges searching area on the subsequent frames and simultaneously estimate a lane model to follow the road boundaries. Lane tracking system has three stages, that is, possible edge scanning, B-spline lane modeling, and Kalman filtering. The nearest edges to the estimated line are determined with the possible edge scanning. A B-spline lane model is constructed with three control points to estimate lane edges in the current frame. Kalman filter corrects the lane model's control points and use them to predict the next frame lane edges. These predicted lane edges will be passed to the next frame for edge scanning again, and this will reduce the overall lane detection process.

3.1. Possible Edge Scanning. The detected lines are used to scan for the nearby edge pixels on the consecutive image after an edge detector, and these lines would be updated iteratively from the previously estimated lane model. The closest pixels to the estimated lane model are considered as the possible

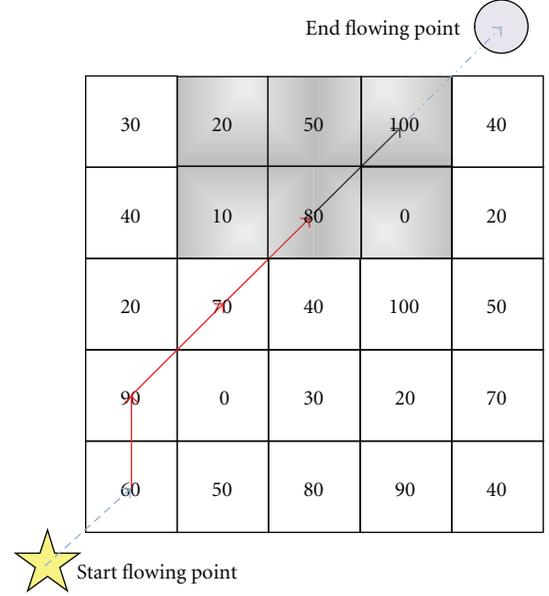


FIGURE 7: The example of flowing path on the edge pixels, where the “star” indicates the starting point; the “circle” indicates the ending point.

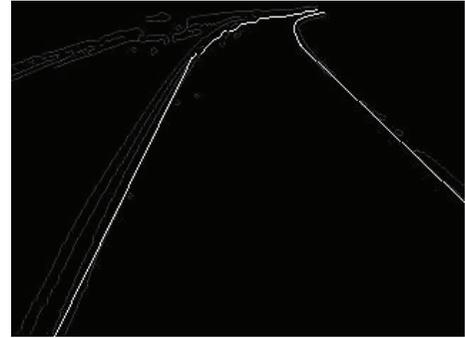


FIGURE 8: The detected edges after river flow model.

lane edges. The detected possible edges are very important for the lane model estimation.

3.2. B-Spline Lane Modeling. An open cubic B-spline [28] with $t + 1$ control points $\{Q_0, Q_1, \dots, Q_t\}$ consists of $(t - 2)$ th connected curve segments, where $t \geq 3$. Each curve segment $g_i(s)$ is a linear combination of four control points, and it can be expressed as

$$g_i(s) = S_i W_i Q_i, \quad i = -1, 0, 1, 2, \dots, t,$$

$$S_i = \begin{bmatrix} s^3 & s^2 & s & 1 \end{bmatrix}, \quad 0 \leq s \leq 1,$$

$$W_i = \begin{bmatrix} w_1 & w_2 & w_3 & w_4 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 0 & 3 & 0 \\ 1 & 4 & 1 & 0 \end{bmatrix}, \quad (6)$$

$$Q_i = \begin{bmatrix} Q_{i-1} & Q_i & Q_{i+1} & Q_{i+2} \end{bmatrix}^T,$$

where S_i is the knot vector which is uniformly distributed from 0 to 1, W_i is the spline basis functions, and Q_i is the control points with 4×4 dimension. According to [24], three control points are found to be efficient to describe the lane shapes, that is, Q_0 , Q_1 , and Q_2 . The Q_0 and Q_2 are the first and last points of the detected edges, while the Q_1 is defined as follows

$$Q_1 = \frac{3}{2}P_1 - \frac{1}{4}(Q_0 + Q_2), \quad (7)$$

where P_1 is the adaptive line segment point. Next, Q_0 and Q_2 are tripled to ensure the line completely passing through the control points. For further prediction, control points have to be rearranged into state \mathbf{x} and the observation model matrix \mathbf{H} :

$$\mathbf{x} = [Q_0 \quad Q_1 \quad Q_2]^T, \quad (8)$$

$$\mathbf{H} = \begin{bmatrix} S_{-1}(w_1 + w_2 + w_3) & S_{-1}w_4 & \mathbf{0} \\ S_0(w_1 + w_2) & S_0w_3 & S_0w_4 \\ S_1w_1 & S_1w_2 & S_1(w_3 + w_4) \\ \mathbf{0} & S_2w_1 & S_2(w_2 + w_3 + w_4) \end{bmatrix}. \quad (9)$$

3.3. Kalman Filtering. Kalman filtering method [6] can be used to predict the control points for left-right edges in consecutive frames. The linear state and measurement equations are defined as

$$\begin{aligned} \mathbf{x}_{k|k-1} &= \mathbf{F}_{k|k-1}\mathbf{x}_{k-1|k-1} + \boldsymbol{\varphi}_k, \\ \tilde{\mathbf{y}}_k &= \mathbf{H}_k\mathbf{x}_{k|k-1} + \mathbf{e}_k, \end{aligned} \quad (10)$$

where the state space \mathbf{x} is the control points of B-spline lane model defined in (8); $\mathbf{F}_{k|k-1}$ is the transition matrix bringing state \mathbf{x} from time $k-1$ to k ; $\boldsymbol{\varphi}_k$ is known as process noise; $\tilde{\mathbf{y}}_k$ is the measurement output; \mathbf{H}_k is the observation model that maps the true state space to the observed space; \mathbf{e}_k is the measurement noise. In this context, $\mathbf{F}_{k|k+1} = \mathbf{I}_{3 \times 3}$ with the assumption of zero external forces. The state is then corrected using Kalman filtering as defined in the following

$$\begin{aligned} \mathbf{P}_{k|k-1} &= \mathbf{F}_{k|k-1}\mathbf{P}_{k-1|k-1}\mathbf{F}_{k|k-1}^T + \mathbf{v}_k, \\ \boldsymbol{\alpha}_k &= \mathbf{y}_k - \mathbf{H}_k\mathbf{x}_{k|k-1}, \\ \mathbf{K}_k &= \mathbf{P}_{k|k-1}\mathbf{H}_k(\mathbf{H}_k\mathbf{P}_{k|k-1}\mathbf{H}_k^T + \mathbf{z}_k)^{-1}, \\ \mathbf{x}_{k|k} &= \mathbf{x}_{k|k-1} + \mathbf{K}_k\boldsymbol{\alpha}_k, \\ \mathbf{P}_{k|k} &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_{k|k-1}, \end{aligned} \quad (11)$$

where $\mathbf{P}_{k|k-1}$ is a priori estimate error covariance; $\mathbf{P}_{k|k}$ is a posteriori estimate error covariance; $\boldsymbol{\alpha}_k$ is the error between the output \mathbf{y}_k obtained from the possible edge scanning and the lane model; \mathbf{K}_k is the Kalman gain. Hence, Figure 9 shows the Kalman filtering-based B-spline lane model which will be used for detecting the possible edges in the next frame.

4. Simulation Results

All results were generated using Matlab 2007a in a machine with core-2 duo processor at 1.8 GHz with 1 GB RAM. An

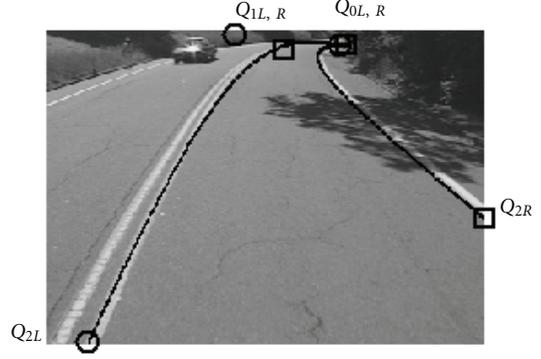


FIGURE 9: The Kalman filtering-based B-spline lane model with control points indication.

image sequence was downloaded from [29], and it was used to compare with the proposed method and the method [24] in terms of performance and computational complexity. Additional video sequences were captured for further experimental evaluation on the proposed lane detection and tracking method. The threshold values of proposed system were initialized as $\varepsilon_{Hz} = 7$, $\lambda = \kappa = 30$, and $T_2 = 10$, where all values were obtained throughout the experiment.

4.1. Performance Evaluation. The performance of the proposed system was evaluated by comparing the estimated lane model with regard to the nearest edges. To calculate an average pixel error, $d = 100$ random points were picked from the estimated lane model, and hence, the error was measured as follows:

$$\text{err} = \frac{\sum |\bar{x}_j - x_j| + |\bar{y}_j - y_j|}{d}, \quad j = 1, \dots, d, \quad (12)$$

where (\bar{x}_j, \bar{y}_j) is denoted as the coordinate of nearest edges, and (x_j, y_j) was the estimated lane model pixel. Figure 10 showed average error plots for left-right edges. The proposed method obtained lower average pixels error rate per frame which were 3.34 and 2.19 for left-right lane tracking than the method [24], which were 7.30 and 6.05, respectively. This was because the performance of method [24] was highly dependent on the CHEVP detection and the threshold value to terminate the measurement of ΔQ and Δk . Examples of using method [19, 24] with its limitations were pointed in Figure 11.

Some lane detection results were provided in Figure 12 where the first line indicated the horizon, and the second line was the near-far-field adaptive line. Figure 12(a) obtained continuous lines on left and right edges, while Figure 12(b) contained dashed lines as the lane marks. The white line was the successfully detected lane edges on the ground using the proposed lane detection method.

Figure 13 showed the simulation results for the proposed method where all continuous frames were detected and tracked successfully. Moreover, more tested results using the proposed method were demonstrated in Figure 14 with two on-road videos. Video no. 1 was recorded in the rural

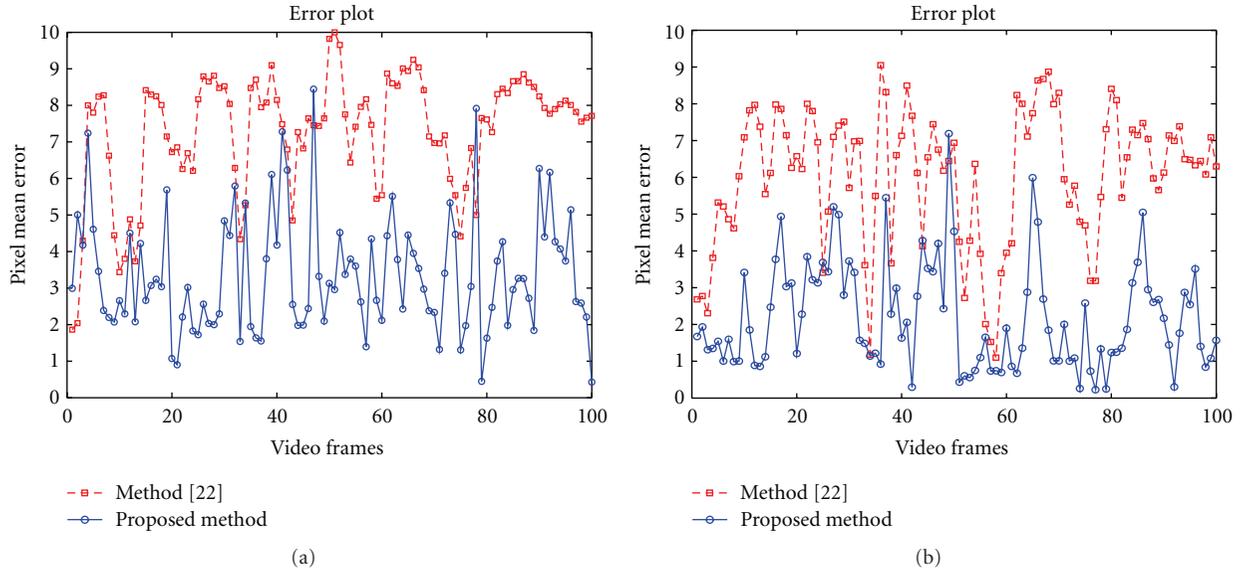


FIGURE 10: The error plots for (a) left lane tracking, (b) right lane tracking.

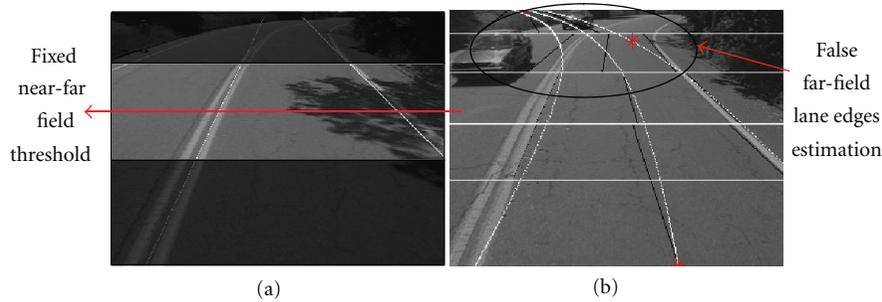


FIGURE 11: Examples of using methods (a) [19] and (b) [24].

area with continuous lines, and it was successfully detected although there was an existing frontal vehicle. Video no. 2 was shot in the highway with dotted dash line as shown in Figure 14(b), and the line was successfully estimated. A demo video has been uploaded to [30] for further understanding. However, the proposed method may still have a failure case in some aspect. When there was an overtaking vehicle which blocks the frontal road edge, it might not detect and predict the lane edges. Massive traffic flow might cause the misdetection too. At the same time, the proposed system may suffer on the scene without lane marks.

4.2. Computational Time. Complexity-wise, the proposed method achieved faster computational time than the method [24]. This was because the complexity of method [24] was greatly dependent on the number of detected lines in the HT process for computing the vanishing points and the GVF iterations which was more complex compared to the proposed method. A summary of computational time between the proposed method and the method [24] was presented in Table 1.

TABLE 1: Complexity comparison in average time base (sec.).

Method	Lane detection	Lane tracking
Method [24]	<4	<0.5
Proposed method	Horizon localization	0.19
	Lane region analysis	0.34
	Adaptive line segment	0.79
	River flow method	0.06

5. Conclusion

A river flow lane detection and Kalman filtering-based B-spline tracking system has been presented to identify lane boundaries in image sequence. The advantages of horizon localization are to limit the searching region on ground and get rid of noises from the sky region. Moreover, lane region analysis eliminates shadow while maintains the lane markings. Meanwhile, an adaptive line segment with multiple EDFs is proposed to monitor the change of lane orientation from near to far-field. A 1D HT is applied to estimate the

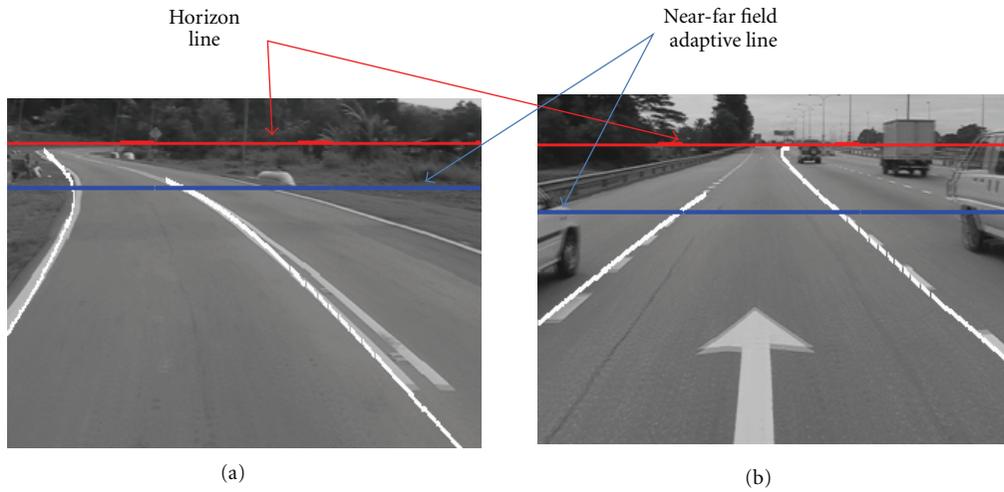


FIGURE 12: The results for the proposed lane detection on (a) rural area, (b) highway.

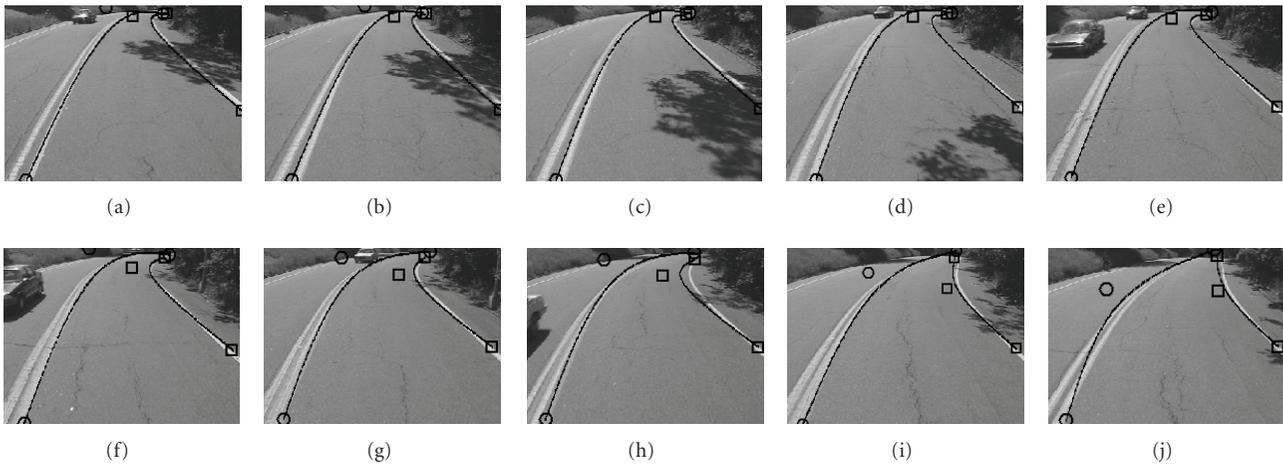


FIGURE 13: Continuous images with the proposed tracking method, where “○”s and “□”s are left-right control points. (c)–(e) Q_{IL} is located outside the image.

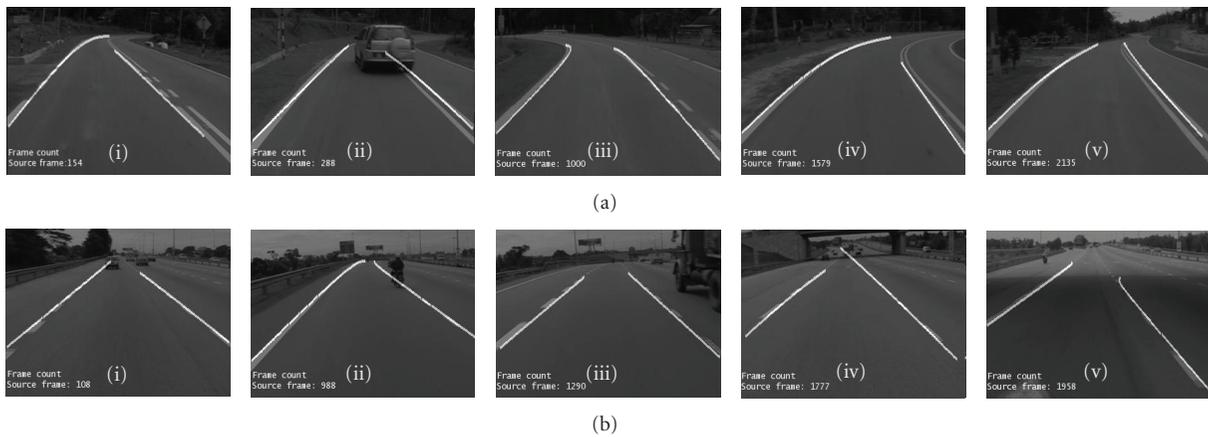


FIGURE 14: Random video samples extracted from the UNMC-VIER AutoVision [27] video clips: (a) video no. 1: rural area and (b) video no. 2: highway.

linear model in the near field. At the same time, river flow model is presented to detect far-field edges, and it could continuously detect and track lane edges for the following frame in future. Finally, B-spline model is predicted with a Kalman filter to follow the lane boundaries continuously. The proposed lane detection and tracking system will be further improved to suit more road scenarios. Further evaluation on river flow method will be investigated in the future. The concept of river flow method could be further extended to attain macroscopic and microscopic optimized mathematical mobility models in the future.

References

- [1] L. Li and F. Y. Wang, *Advanced Motion Control and Sensing for Intelligent Vehicles*, Springer, New York, NY, USA, 2007.
- [2] J. M. Armingol, A. de la Escalera, C. Hilario et al., "IVVI: intelligent vehicle based on visual information," *Robotics and Autonomous Systems*, vol. 55, no. 12, pp. 904–916, 2007.
- [3] Y. Zhou, R. Xu, X. F. Hu, and Q. T. Ye, "A robust lane detection and tracking method based on computer vision," *Measurement Science and Technology*, vol. 17, no. 4, pp. 736–745, 2006.
- [4] M. Bertozzi, A. Broggi, M. Cellario, A. Fascioli, P. Lombardi, and M. Porta, "Artificial vision in road vehicles," *Proceedings of the IEEE*, vol. 90, no. 7, pp. 1258–1270, 2002.
- [5] V. Kastrinaki, M. Zervakis, and K. Kalaitzakis, "A survey of video processing techniques for traffic applications," *Image and Vision Computing*, vol. 21, no. 4, pp. 359–381, 2003.
- [6] J. C. McCall and M. M. Trivedi, "Video-based lane estimation and tracking for driver assistance: survey, system, and evaluation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 20–37, 2006.
- [7] A. Bar Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine Vision and Applications*. In press.
- [8] Y. Wang, N. Dahnoun, and A. Achim, "A novel system for robust lane detection and tracking," *Signal Processing*, vol. 92, no. 2, pp. 319–334, 2012.
- [9] J. D. Crisman and C. E. Thorpe, "SCARF. A color vision system that tracks roads and intersections," *IEEE Transactions on Robotics and Automation*, vol. 9, no. 1, pp. 49–58, 1993.
- [10] M. A. Turk, D. G. Morgenthaler, K. D. Gremban, and M. Marra, "VITS—a vision system for autonomous land vehicle navigation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 342–361, 1988.
- [11] K. Kluge and C. Thorpe, "The YARF system for vision-based road following," *Mathematical and Computer Modelling*, vol. 22, no. 4–7, pp. 213–233, 1995.
- [12] Z. Kim, "Robust lane detection and tracking in challenging scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 1, pp. 16–26, 2008.
- [13] S. Baluja, "Evolution of an artificial neural network based autonomous land vehicle controller," *IEEE Transactions on Systems, Man, and Cybernetics B*, vol. 26, no. 3, pp. 450–463, 1996.
- [14] C. Thorpe, M. H. Hebert, T. Kanade, and S. A. Shafer, "Vision and navigation for the Carnegie-Mellon navlab," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, no. 3, pp. 362–373, 1988.
- [15] D. Pomerleau, "RALPH: rapidly adapting lateral position handler," in *Proceedings of the Intelligent Vehicles Symposium*, pp. 506–511, September 1995.
- [16] S. K. Kenue and S. Bajpayee, "LaneLok: robust line and curve fitting of lane boundaries," in *Proceedings of the 7th Mobile Robots*, pp. 491–503, Boston, Mass, USA, November 1992.
- [17] D. J. LeBlanc, G. E. Johnson, P. J. T. Venhovens et al., "CAPC: a road-departure prevention system," *IEEE Control Systems Magazine*, vol. 16, no. 6, pp. 61–71, 1996.
- [18] C. Kreucher and S. Lakshmanan, "LANA: a lane extraction algorithm that uses frequency domain features," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 343–350, 1999.
- [19] C. R. Jung and C. R. Kelber, "Lane following and lane departure using a linear-parabolic model," *Image and Vision Computing*, vol. 23, no. 13, pp. 1192–1202, 2005.
- [20] C. R. Jung and C. R. Kelber, "An improved linear-parabolic model for lane following and curve detection," in *Proceedings of the 18th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI'05)*, pp. 131–138, October 2005.
- [21] Y. Wang, L. Bai, and M. Fairhurst, "Robust road modeling and tracking using condensation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 9, no. 4, pp. 570–579, 2008.
- [22] L. Bai, Y. Wang, and M. Fairhurst, "An extended hyperbola model for road tracking for video-based personal navigation," *Knowledge-Based Systems*, vol. 21, no. 3, pp. 265–272, 2008.
- [23] Y. Wang, D. Shen, and E. K. Teoh, "Lane detection using spline model," *Pattern Recognition Letters*, vol. 21, no. 8, pp. 677–689, 2000.
- [24] Y. Wang, E. K. Teoh, and D. Shen, "Lane detection and tracking using B-Snake," *Image and Vision Computing*, vol. 22, no. 4, pp. 269–280, 2004.
- [25] C. Y. Xu and J. L. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.
- [26] K. H. Lim, K. P. Seng, and L. M. Ang, "Improvement of lane marks extraction technique under different road conditions," in *Proceedings of the 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT'10)*, pp. 80–84, Chengdu, China, July 2010.
- [27] K. H. Lim, A. C. Le Ngo, K. P. Seng, and L.-M. Ang, "UNMC-VIER AutoVision database," in *Proceedings of the International Conference on Computer Applications and Industrial Electronics (ICCAIE '10)*, pp. 650–654, Kuala Lumpur, Malaysia, December 2010.
- [28] K. I. Joy, "Cubic uniform B-spline curve refinement," in *On-Line Geometric Modeling Notes*, University of California (Davis), 1996.
- [29] Carnegie-Mellon-University, "CMU/VASC image database, 1997–2003," http://vasc.ri.cmu.edu/idb/html/road/may30_90/index.html.
- [30] K. H. Lim, "River flow lane detection and Kalman filtering based B-spline lane tracking," 2011.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

